RICOH Web Enablement Solutions Suite: Summary of Updates 1.3

The main updates for Web Enablement Solutions Suite, version 1.3, are summarized in the table below.

Date	Release number	Summary of changes
2020.03.10	1.300.00	**Web Enablement Solutions Suite 1.3 Release**
2020.03.20	1.300.01	**DE38227** - Fixed the processing issue for the Set Intercharacter Adjustment, when Set Variable Space Character Increment is also used.
2020.03.30	1.300.02	**DE38243** - Fixed the missing characters issue when using Type1 Japanese fonts.
2020.04.03	1.300.03	**DE38246** - Fixed the issue regarding the GOCA pattern change from the options file. The change also extends to another GOCA pattern.
2020.04.07	1.300.04	**DE38243** - Fixed customer issue related to incorrect width for some characters and a glyph that did not appear as expected. This new fix addresses both the original 1.300.02 fix and the new issue.
2020.04.15	1.300.05	**DE38408** - Created a special case for code page 300 to use a custom charset instead of the default Java one, because of inconsistent output when decoding arrays that contain unmappable code points. The custom charset has the same mapping as the standard Java charset for code page 300, but the decoding algorithm is different.
2020.05.05	1.300.06	**DE38530** - Fixed client code to handle the End of File marker properly.
2020.05.07	1.300.07	**DE38612** - Fixed alignment when Set Intercharacter Adjustment is used with horizontal scaling (width override) from `csdef.fnt` or when the character rotation is used.
2020.05.19	1.300.08	**US53549** - Added page-piece dictionary to PDF when the odload option is used with the ArchiveLoad_afp2pdf command for loading CMOD.
2020.05.29	1.300.09	**DE38788** - Changed TLE processing for an AVE compatibility issue.
2020.06.18	1.300.10	**DE38664** - Added the option to select the legacy processing for: • The formula to convert from CMYK to RGB for images. • Font map processing.
2020.07.01	1.300.11	**US53650** & **US53667** - Ricoh internal fixes.
2020.07.02	1.300.12	**DE39005** - Fixed issue with additional invalid characters added by the transform, when a UDC section is missing.

2020.07.04	1.300.13	**DE39004** - Use an IBM Shift-JIS encoding when creating strings from a SHIFT-JIS code page mapping file.
2020.07.21	1.300.14	**DE39131**: • Modified position calculation for Type3 glyphs. • Added *Type3Glyph_Position_Legacy*, a new options file parameter. **DE39192** - Fixed an error causing the last character of mixed UDC and non-UDC TRN text to be lost.
2020.08.07	1.300.15	**DE39293** - Fixed various font issues.
2020.08.13	1.300.16	**US54779** - When text characters cannot be mapped in the requested mapped font, use the AFP Font to convert those characters to image.
2020.08.28	1.300.17	**DE39460** - Moved horizontal scale calculation to avoid being used twice when SIA is applied. **DE39476** - Used vertical encoding for Chinese, Japanese, and Korean (CJK) fonts, when the character rotation is 90 or 270.
2020.09.08	1.300.18	**US55110** - Added support for double values in the image map configuration file.
2020.09.14	1.300.19	**DE39520** - Added handling for an iText exception thrown for inputs with incorrect or unmatching AFP font resources.
2020.10.08	1.300.20	**DE39735**: • Form definition pointer not to be used if on internal medium map. • ASCII space not to be used as substitute for code point 0 for type3 fonts.
2020.10.15	1.300.21	**DE39821** - When SBCS text converts to contain either unicode replacement character u+fffd or u+001a, replace with a blank character, when no font is available for imaging.
2020.10.20	1.300.22	**DE39883** - Fixed IOCA mask processing for double-dotted images.
2020.10.22	1.300.23	**DE39860** - When using image mapping, place static images and shaded areas ahead of medium overlays.
2020.10.30	1.300.24	**US55365** - Added new option to allow TrueType Font processing to search for `.ucm` files prior to search for `.cp` files.
2020.11.03	1.300.25	**DE39960**: • Fixed underscore processing to allow an underscore control to remain active across multiple TRN text controls. • Added support for overstrike text controls.
2020.11.13	1.300.26	**DE40046** - Remove white space from each line while processing various *.fnt files. **DE40047** - Remove double quotes while reading passwords from the options file.

2020.11.17	1.300.27	**DE40049** - Fixed clipping and rounding issues for rotated RGB images.
2020.11.18	1.300.28	**DE40112**: • Fixed IOCA single tile with tile offset issue. • Fixed image mapping issue with page extent.
2020.11.23	1.300.29	**US53704**: • Redesign the processing of object containers with non MODCA images. • Fixed object area size not being correctly inherited from parent when it was not present in either IOB or container OEG. • Added TIFF images to cache similar to how JPEGs were handled. • Added unit base processing (cm vs inch) where missing. **US55966**: • Added support for image resolution, with and without triplet 9A for non MODCA images. • Fixed CDD triplet parsing.
2020.12.01	1.300.30	**DE40153** - Allowed user offset to apply to image map files.
2020.12.08	1.300.31	**DE40272** - Saved clipping size in cached page segments so it can be restored on each use and across job boundaries.
2020.12.14	1.300.32	**US40491** - Added support for non MODCA images placed directly on page/overlay.
2021.01.07	1.300.33	**DE40441** - Ignored empty option file entries.
2021.01.12	1.300.34	**DE40485** - Converted single byte section number to integer by ANDing it with 0xff to keep the value as a positive number; section numbers are never negative.
2021.01.13	1.300.35	**DE40508** - Set Intercharacter Adjustment needs to be ignored for blank characters.
2021.01.22	1.300.36	**DE40533** - Improved memory usage for image mapping jobs that use only *static* type entries; only by adding, not by replacing.
2021.01.28	1.300.37	**DE40571**: • Print check digit in HRI when bar code symbology/modifier specifies it is to be printed. • Use the correct font mapping files for the HRI font. • Use the correct value for the check digit in the bar code symbol generated.
2021.02.02	1.300.38	**US57162** - Added support for multi-page TIFF images.
2021.02.08	1.300.39	**US57387**: • Added support for Windows Server 2019. • Enhanced server start/stop capabilities.

2021.02.09	1.300.40	**US57217** - Upgraded the Bouncy Castle libraries to version 1.68.
2021.02.15	1.300.41	**US57319**: • Added a new filter to set the wide-to-narrow ratio for Code39 and Int. 2 of 5 bar code objects to match AFP2PDF native. • Mapped HRI fonts to use the same font as the text.
2021.02.18	1.300.42	**US57443** - Added the a2pxopts configuration option for PDF Producer.
2021.03.03	1.300.43	**DE40679** - Replaced code point 0 with an empty character when mapping fonts.
2021.03.10	1.300.44	**DE40869** - Corrected the code point for the hyphen character within Type 1 fonts.
2021.03.18	1.300.45	**US58161** - Added support for external resource groups and external form definitions for all AFP Visual Environment filters.
2021.04.01	1.300.46	**DE41031** - For no name on Font Descriptor, use character set name.
2021.04.12	1.300.47	**US55382**: • Added the functionality of logging to separate files in server and API mode. • Increased default cache size for resources to 100M.
2021.04.28	1.300.48	**US58692** - Improved font bounding box calculations.
2021.05.05	1.300.49	**DE41186**- Fixed the TrueType font mapping to allow mapping for horizontal and vertical writing modes.
2021.05.14	1.300.50	**DE41301** - Set default character as blank when marked as invalid.
2021.05.19	1.300.51	**DE41358** - Use default code point or blank character for undefined code points when converting to Type 3 fonts.
2021.05.27	1.300.52	**DE41380** - Ignore the `IOCA Subsampling` parameter when the total amount of image data indicates no subsampling.
2021.06.03	1.300.53	**US42265** - Added the `Capture_Files_for_Debug` parameter to the options file to capture all debugging files within a zip file.
2021.06.28	1.300.54	**DE41544** - Fixed the Constant Forms processing for basic N-UP/no N-UP using default page placement.
2021.07.07	1.300.55	**DE41650** - Added processing of Horizontal Scale Factor for Data Object fonts (TTF, OTF).
2021.07.13	1.300.56	**DE41699** & **DE41663**: • Fixed processing of external form definition in filters for non-existing inline resource group. • Fixed `split_afp2pdf` to issue a warning message for duplicated file names when using the `legacylog` flag. • Fixed the `ignore_bar_code_object_area` parameter to be ignored when checking if the image exceeds the page size.

2021.07.23	1.300.57	**DE41749** - Updated internal codepage 837 with newer `.ucm` file.
2021.07.26	1.300.58	**DE41669**, **DE41741** & **DE41749.2**: • Revised the processing of PDF objects. • Converted EBCDIC QR Code data to codepage 897 with the use of internal tables. • Fixed **DE41749** to also work with the IBM J9 Virtual Machine (JVM).
2021.07.30	1.300.59	**DE41716** & **DE41663.2**: • Changed the Codabar minimum module width to match the legacy AFP2PDF and corrected the symbol width calculation. • Changed `split_afp2pdf` to only issue duplicated file warning messages from one thread.
2021.08.13	1.300.60	**DE41870**: • Fixed non-MODCA image objects processing of the position mapping option to not convert when the image exceeds the object area. • Fixed processing of TIFF to allow random access instead of sequential.
2021.08.23	1.300.61	**DE54098** - Ignore the `-host` parameter. The use of this parameter is allowed only for backward compatibility purposes.
2021.08.24	1.300.62	Internal use only
2021.09.02	1.300.63	**DE42037** & **US42518**: • Fixed high memory usage when processing PDF resources. • Increased maximum heap size in start server script.
2021.09.13	1.300.64	**DE42061** - Fixed font caching to allow font with same name but different content. **DE42067** - Fixed TIFF processing to allow different type of TIFF image. **US61289** - If AFP codepage does not contain Unicode values, try using the mapped codepage info to obtain Unicode value. **US56252** - Added new parameter to allow for PDF output pages to be resized to contain offsets from form definition and options file.
2021.09.21	1.300.65	**DE42158** - Fixed G4 Decompression error on IOCA image.
2021.09.28	1.300.66	**US54098.2** - Fixed the client code for z/OS USS and Linux on z broken by 1.300.61.
2021.10.13	1.300.67	**DE42239 & DE42240** - Processed an empty TRN properly. Processed font with incorrect default character.

2021.10.15	1.300.68	**US61887** - Added the option to install AFP2PDF Plus server as a service on supported platforms (Windows, Aix , Linux and zLinux).
2021.10.20	1.300.69	**DE42341** - Fixed log files ignoring size limit from logging. properties
2021.11.04	1.300.70	**US61628** - Added support for vertical text when using TTF fonts and when using mapped TTF fonts.
2021.11.15	1.300.71	**DE42504** - Make sure the start/stop server scripts have the shebang entry (#!/bin/sh) when upgrading from an older version of Plus.
2021.12.08	1.300.72	**DE42645 & DE42646**: • Fixed the EAN8 barcode: incorrect middle guard character drawing. • Fixed the Codabar (NW7) barcode: incorrect checkdigit position and special case for 0. • Fixed the PDF417 barcode: element height not using the correct BCOCA unit. • Fixed the Code39 barcode: symbol width calculation not taking into account intercharacter gaps.
2021.12.15	1.300.73	**DE42654 & DE42655 & DE42656**: • Fixed nullpointerexception when TIFF resource not found AND fixed TIFF decompression error. • Fixed char rotation 90 positioning. Use AFP codepage when specified for barcode HRI.
2021.12.22	1.300.74	**DE42654 continued** - When searching for external non-MODCA resources (tif, jpg,gif, pdf), we are also using file type specific extensions. **US61976** - Updated Apache Santuario (xmlsec) library to version 2.2.3 to fix security vulnerability.
2022.01.13	1.300.75	**WE3-459**: • Fixed inconsistent error codes when having memory usage issues with split_afp2pdf. • Added page segment caching for page segments using the LDOPM triplet.
2022.01.21	1.300.76	**WE3-454 (DE42655.2)** - Fixed char rotation 90 positioning.
2022.01.21	1.300.77	**WE3-473** - Fixed split_afp2pdf external resource issue caused by misused static member in multithreaded environment.
2022.02.04	1.300.78	**WE3-475** - Fixed Archive_load sometimes not showing content from page segment resources.
2022.02.08	1.300.79	**WE3-478** - Fixed an iText issue that was causing an exception with non-pdf output when using pfb resources that did not have a certain format.
2022.02.15	1.300.80	**WE3-454** - Additional fix for positioning of rotated characters.

2022.02.25	1.300.81	Removed a smallportion of logging information that could cause crashes on Java versions starting with 9
2022.03.10	1.300.82	**WE3-486** Improve handling of badly formatted jpeg images: instead of terminating the entire processing without any pdf ouput, just skip the bad image, log it and continue processing.
2022.03.28	1.300.83	**WESS-3413** Improve accuracy of relative and absolute movement in PTOCA. WESS-901 Improving caching of non MODCA image resources and PDF objects by adding a configurable size limit.
2022.05.04	1.300.84	**WE3-492** Modified the way TrueType fonts are subsetted so that they work with custom code-page (1038 - symbol cp).
2022.05.23	1.300.85	**WE3-497** Fixed a crash caused by missing IOCA resources.
2022.06.07	1.300.86	**WE3-499** Fixed an issue that was causing external resources groups passed as parameters to be in use even after their respective job was done, making it so they could not be deleted while the server was running.
2022.06.07	1.300.87	**WE3-500** Fixed processing of default character flags for AFP code page/charset resources.
2022.06.10	1.300.88	**WESS-3553** Fixed split_afp2pdf XML output when using the *-tle* option: escaped special XML characters in the *tle* name and value.
2022.08.11	1.300.89	**WESS-3634** Add support in the installer for Windows 11 and Windows Server 2022.
2022.09.09	1.300.90	**WE3-508** Fixed a multithread related issue that could cause caching of page segments in server mode to lose content, due to it being clipped out by a 0 bounding box. **WE3-506** Update User Guide description of the *-tle* parameter for split_afp2pdf.
2022.10.10	1.300.91	**RPD3-5200** AFPParser issue from VWB - fixed PTX issue related to chained control sequences.
2022.10.20	1.300.92	**WE3-504** Ensured that the font-path is consistent, so that it is processed only in the first thread.
2022.11.07	1.300.93	**WE3-504** Synchronized setting the decoder for TrueType fonts. This prevents potential errors when running multiple threads in split or server mode.
2022.11.17	1.300.94	**WESS-3658** Updated the communication with PDF2PDFUA; **WESS-3718** Added the Disable_TTF_Subsetting config option.
2023.01.10	1.300.95	**WE3-521** Commit changes for adding `imagemap` support for AFP2PDF Split.
2023.02.06	1.300.96	**WESS-702** Add support for rotated IPO; WESS-218 - Add support for SEC CIELab and Highlight color space.

2023.02.12	1.300.97	**WE3-520** Improve support for JPEG compressed images using Apache Commons Imaging when our regular image processing fails.
2023.02.13	1.300.98	**WE3-527** Added a limitation to the size of the glyph bounding box for Type3 fonts to remove Acrobat Reader error.
2023.02.15	1.300.99	**WE3-528** Updated ImageGenerators.jar to support RGBs with alpha-channel.
2023.02.23	1.301.01	**WE3-529** Added default basline increment value.
2023.02.24	1.301.02	**WE3-530** Fixed a PGP repeating group index that was adding a PMC overlay on a blank page.
2023.03.13	1.301.03	**WESS-3929** Added new config file option to output the number of transfomed pages.
2023.03.27	1.301.04	**WESS-4085** Improve image mapping to allow the addition of an alternate text for images. This is only relevant when using Afp2pdf Plus in conjuction with the PDFUA tool.
2023.04.03	1.301.05	**WESS-3929** Add OpenTypoe and TrueType font mapping support.
2023.04.12	1.301.06	**WESS-4064** Updated copyright in source-code.
2023.04.27	1.301.07	**WESS-3605** Fix Extended Bilevel Image Color in IOCA.
2023.05.26	1.301.08	**WE3-535** Remove redundant clipping from GOCA objects that was causing unwanted behavior (missing content) when page offsets were specified in the options file.A89
2023.05.30	1.301.09	**WE3-533** Added new command line option (-c) to AFP2PDF Split to specify the size of the cache. WESS-4300 - Fix a rare issue that could cause flag parameters (like -m for example) to be ignored in server mode if they were the last one. WESS-4301 - Fix a null pointer exception when using some unsupported but tolerated parameters (-l, -g, -s).
2023.06.09	1.301.10	**WE3-536** Fixed a typg o that was preventing mapping to Helvetica-Narrow type fonts; Fixed a font mapping issue that in certain situations would cause mappinto Adobe base fonts to be ignored.
2023.06.19	1.301.11	**WESS-4362** Fixed a crash when calling the transform with an option file without any valid options inside. **WESS-4307** Integarated ImageGenerator code into afp2pdf plus projects. **WE3-538** Fixed some scripts not correctly passing arguments with spaces inside.
2023.07.03	1.301.12	**WESS-4377** Fixed the fordef parameter that was not working for split_afp2pdf when image mapping was not used. Removed JAXB dependency from Afp2pdfplit so it works with newer versions of the JVM that no longer ship this package.

2023.07.10	1.301.13	**WESS-4236** Implemented native `DASH_PATTERN_style` configuration file option in AFP2PDF Plus.
2023.07.23	1.301.14	**WE3-540** Added additional check for DBCS Raster code-pages when matching code-pages from `icoded` with the ones from `cpdef`.
2023.08.03	1.301.15	**WESS-4431** Improved processing of AVE control files to allow "pretty print" formatted XMLs.
2023.08.21	1.301.16	**WESS-4442** Improved support for running AFP2PDF Plus as a service on Windows. Added a sleep equivalent during uninstall to prevent IA from stalling.
2023.08.30	1.301.17	**WESS-4236** Fix for handling configuration entries while processing dash line patterns when some entries are commented out.
2023.10.03	1.301.18	**WE3-551** Fix TTF processing to tolerate fonts that use the Unicode Supplementary Private Use Areas without having an alternate code in a non private use area. **WE3-550** Page content is using page position when processing a 3-UP file, although it should not. Added `STATIC_PAGE_LENGTH` configuration file option to specify the logical page size.
2023.10.05	1.301.19	**WESS-4586** Update default encryption for password protected pdfs to AES256. Update Datalogics PDF Java Toolkit to 5.0.7. **WESS-4616** Update Bouncy Castle libraries to 1.76 to fix security vulnerabilities.
2023.10.12	1.301.20	**WE3-552** Fix an unhandeled exception that could occur in very niche cases involving input files with PDF417 barcodes.
2023.11.03	1.301.21	**WESS-4639** Made a change that makes the order some GOCA objects are written more consistent across different OSs and/or JVMs. The change has no impact on output size or appearance, it was just needed for our regression tests. **WESS-4656** Added support for GS1 AI encodable character set 39 and 82.
2023.12.11	1.301.22	**WESS-4313** Add global scale command line option. Fixed a split_afp2pdf issue that could prevent outputs to be generated if the output names were based on TLE information containing special characters (backslash).
2024.01.23	1.301.23	**WESS-4828** Updated Apache xmlsec library to version 2.2.6. **WESS-4827** Migrated build process to use Install Anywhere 2022 when creating the final installation package.
2024.01.29	1.301.24	**WE3-557** Updated the GOCA Set current defaults instruction to take in account the default bit in byte 5.

2024.01.31	1.301.25	**WE3-556** Fixed performance issue in server and API mode caused by unnecessary overhead in the caching mechanism for page segment and overlay resources.
2024.03.21	1.301.26	**WE3-564** Made changes so that the logging option specified via API overrides the configuration file option.
2024.04.03	1.301.27	**WE3-564** Added flushing of the logging handler so that only warnings work. **WESS-4924** Fixed null pointer in ImageGenerators found while test IOB rotation for AFP Visual Environment.
2024.04.15	1.301.28	**WE3-567** Fixed a crash when processing True Type fonts without a glyph substitution map.
2024.04.16	1.301.29	**RPD3-5775** Made changes to the AFPParser to make it more thread-safe, removing a potential deadlock during AFPSf initialization.
2024.04.24	1.301.30	**WESS-4521** RICOH internal changes.
2024.05.10	1.301.31	**WE3-569** Memory usage improvements related to image caching including a new options file entry: `PER_JOB_IMAGE_CACHE`.
2024.05.24	1.301.31	**WESS-5157** Mitigate vulnerability CVE-2017-9096 in iText.
2024.06.07	1.301.33	**WESS-5215** Added support for inflate/deflate compression in TIFF images.
2024.08.06	1.301.34	**WESS-5342** Updated Bouncy Castle to v1.78.1. **WESS-5366** Improved image mapping config file parsing to tolerate some niche cases.
2024.08.09	1.301.35	**WE3-575** Adjusted Type3 glyph bounding box to solve printing issues on some 3rd party printers.
2024.08.14	1.301.36	**WESS-5272** Added support for `USE_AFP_METRICS` config file option that allows to use the AFP fonts metrics on mapped fonts.
2024.09.03	1.301.37	**WESS-4705** Fixed a crash when using `Capture_Files_For_Debug` option file entry in API mode. Added input and output to the debug capture. **WESS-5394** Improved error logging in API mode.
2024.09.13	1.301.38	**WE3-586** Fixed alternate text not being added for images when `Preserve_CMYK` was set to false. **WE3-583** Fixed missing resource log entries not showing for certain types of resources.
2024.09.19	1.301.39	**WE3-587** Fixed a bug that was causing 0 size output in niche scenarios where an external `formdef` was used.
2024.09.26	1.301.40	**WE3-590** Adjusted the fill GOCA patterns for patterns 9 to 13 (filling using vertical, horizontal or diagonal lines).
2024.10.07	1.301.41	**WE3-588** Added new config file option `GOCA_COLOR_OF_MEDIUM` to specify the color of medium.

2024.10.22	1.301.42	**WE3-581** Changed logging when running through API to allocate each thread to its own logger, thus removing OOM error for large number of threads.
2024.10.31	1.301.43	**WE3-594** Fixed Type3 glyph positioning. Recalculations made the `Type3Glyph_Position_Legacy` options file entry useless.
2025.01.31	1.301.45	**WESS-5743** Improved `Disable_Compression` option.
2025.02.03	1.301.46	**WE3-603** Fixed XXE (XML External Entity) vulnerability.
2025.02.11	1.301.47	**WE3-606** Fixed clipping issue with page segments when page offsets are specified in the options file. **WE3-608** Fixed missing content in output when image mapping `INDEX_STATICIMG_...` type options were used with page segment or overlay resources.
2025.02.25	1.301.48	**WESS-5659** Fixed the `Capture_Debug` option not capturing the log file in server mode.
2025.03.26	1.301.49	**WE3-615** Fixed character space increment not being picked up from character set resources that were not named exactly as they were referenced, such as `C0M24F41` vs. `C1M24F41`.
2025.04.10	1.301.50	**WESS-5927** Extend the `-p` parameter functionality to allow specifying a page range (previously only a single page was allowed).
2025.04.22	1.301.51	**WESS-5929** Added a default font configuration choice (mapping vs. embedding fonts) when installing AFP2PDF Plus.
2025.05.22	1.301.52	**WESS-6218** Added more specific logging to errors coming from the **ua** plugin. Fixed potential version out of sync error with the **ua** plugin.